19PRTS
09/284244
510 Rec'd PCT/PTO 14 APR 1999

DESCRIPTION

Method of Processing Data in FM Subcarrier Data Broadcasting Receiver

Technical Field

The present invention relates to a method of processing data in an FM subcarrier data broadcasting receiver.

Background Art

FM subcarrier data broadcasting is one for broadcasting voices, characters, figures, and so forth together with stereo voices. That is, it is one for multiplexing data such as voices, characters, and figures on normal FM broadcasting and broadcasting the multiplexed data.

An FM subcarrier data broadcasting receiver comprises a multiple receiving LSI 22 for performing LMSK demodulation and error correction on the basis of an output from a tuner 21, a receiving processing unit 23 for accepting only required one of data in units of blocks outputted from the LSI 22, a program reconstructing unit 24 for reconstructing a program, a program analyzing unit 25 for performing decoding processing according to an eight bit coding system, a display processing unit 26 for acquiring a drawing pattern as well as subjecting the drawing pattern

to various types of processing, and a drawing unit 27 for outputting the drawing pattern obtained by the display processing unit 26 to a display device 28, as shown in Fig. 24. Processing of the receiving processing unit 23, the program reconstructing unit 24, the program analyzing unit 25, the display processing unit 26, and the drawing unit 27 is performed by a CPU (not shown). That is, it is executed by software.

Fig. 25 shows the configuration of conventional software.

The software comprises a receiving processing portion for acquiring data (layer 3 data) from an FM multiple LSI, a program reconstructing portion for classifying the layer 3 data to construct a data group (layer 4 data), and a program analyzing portion for analyzing (decoding) each of data units (layer 5 data) included in the data group to present information (layer 6 data). The program analyzing portion manages a display processing portion and a drawing portion.

Fig. 26 shows the procedure for processing in a case where data constituting a reconstructed program are displayed.

When a drawing page is designated (step 101),

data in one display unit (for example, data in one character unit) constituting the page are decoded in accordance with an eight bit coding system (step 102). Consequently, a character code or the like is acquired.

A drawing pattern corresponding to the obtained character code or the like is then obtained (step 103). The obtained drawing pattern is subjected to processing such as enlargement, underline addition, and inversion (step 104). The drawing pattern obtained after the processing is drawn in a designated position of a display device (step 105).

When the processing at the steps 102 to 105 is repeated for each data in one display unit, and all data constituting the page are subjected to the processing (YES at step 106), the program is returned to the step 101.

The display data corresponding to one page can be also collectively displayed after being expanded into a drawing memory, which departs from standards for FM subcarrier data broadcasting.

In order to develop software of the FM subcarrier data broadcasting receiver, complicated specifications peculiar to FM subcarrier data broadcasting must be understood. Therefore, much

time and labor are required to develop the receiver.

If the portions which relate to the specifications peculiar to FM subcarrier data broadcasting and the portions which do not relate thereto can be separated from each other, it is considered that the FM subcarrier data broadcasting receiver is easy to develop. In the conventional FM subcarrier data broadcasting receiver, however, the program analyzing unit directly relates even to processing relating to display. Therefore, the portions which relate to the specifications peculiar to FM subcarrier data broadcasting and the portions which do not relate thereto cannot be separated from each other in the configuration of the software.

An object of the present invention is to provide a method of processing data in an FM subcarrier data broadcasting receiver in which portions which relate to specifications peculiar to FM subcarrier data broadcasting and portions which do not relate thereto can be separated from each other in the configuration of software, and an FM subcarrier data broadcasting receiver is easy to develop.

Disclosure of Invention

A method of processing data in an FM subcarrier data broadcasting receiver according to the present

invention is characterized in that in displaying data constituting a reconstructed program, an eight bit code is decoded according to an eight bit coding system, and an intermediate code which can be decoded even if specifications peculiar to FM subcarrier data broadcasting are not understood is generated, to perform display control processing after the intermediate code is decoded.

An intermediate code corresponding to a character includes an intermediate code corresponding to a character having no attribute and an intermediate code corresponding to a character having attributes. The intermediate code corresponding to the character having no attribute comprises a code representing the character having no attribute, a display position, a character size, and a character code, for example, and the intermediate code corresponding to the character having attributes comprises a code representing the character having attributes, a display position, a character size, a character code, a font attribute, and a display attribute, for example.

An intermediate code corresponding to a one-layer photographic includes an intermediate code corresponding to a one-layer photographic having no

attribute and an intermediate code corresponding to a one-layer photographic having an attribute. The intermediate code corresponding to the one-layer photographic having no attribute comprises a code representing the one-layer photographic having no attribute, a display position, a size, and data, for example, and the intermediate code corresponding to the one-layer photographic having an attribute comprises a code representing the one-layer photographic having an attribute, a display position, a size, data, and a display attribute, for example.

An intermediate code corresponding to an external character code set includes an intermediate code corresponding to an external character code set having no attribute and an intermediate code corresponding to an external character code set having attributes. The intermediate code corresponding to the external character code set having no attribute comprises a code representing the external character code set having no attribute, a display position, a character size, and data, for example, and the intermediate code corresponding to the external character code set having attributes comprises a code representing the external character

code set having attributes, a display position, a character size, data, a font attribute, and a display attribute, for example.

An intermediate code corresponding to a geometric drawing instruction includes intermediate codes corresponding to a dot drawing instruction, a straight line drawing instruction, a rectangle drawing instruction, a polygon drawing instruction, and a circle or circular arc drawing instruction. The intermediate code corresponding to each of the drawing instructions includes a code representing the type of the drawing instruction, a pixel size, and coordinate positions required to draw the drawing instruction. All the coordinate positions required to draw the drawing instruction are given by absolute coordinates.

Brief Description of Drawings

Fig. 1 is a block diagram showing the electrical configuration of an FM subcarrier data broadcasting receiver;

Fig. 2 is a flow chart showing the procedure for processing in a case where data constituting a reconstructed program are displayed;

Fig. 3 is a schematic view showing the structure of an intermediate code corresponding to a character

(having no attribute);

Fig. 4 is a schematic view showing the structure of an intermediate code corresponding to a character (having attributes);

Fig. 5 is a schematic view showing the structure of an intermediate code corresponding to a one-layer photographic (having no attribute);

Fig. 6 is a schematic view showing the structure of an intermediate code corresponding to a one-layer photographic (having an attribute);

Fig. 7 is a schematic view showing the structure of an intermediate code corresponding to DRCS (having no attribute);

Fig. 8 is a schematic view showing the structure of an intermediate code corresponding to DRCS (having attributes);

Fig. 9 is a schematic view showing the structure of an intermediate code corresponding to screen erasure;

Fig. 10 is a schematic view showing the structure of an intermediate code corresponding to TIME;

Fig. 11 is a schematic view showing the structure of an intermediate code corresponding to selective control information;

Fig. 12 is a schematic view showing the structure of an intermediate code corresponding to color change;

Fig. 13 is a schematic view showing the structure of an intermediate code corresponding to a color map;

Fig. 14 is a schematic view showing the structure of an intermediate code corresponding to a page attribute;

Fig. 15 is a schematic view showing a coordinate system used in an intermediate code;

Fig. 16 is a schematic view showing the structure of an intermediate code corresponding to a point;

Fig. 17 is a schematic view showing the structure of an intermediate code corresponding to a straight line;

Fig. 18 is a schematic view showing the structure of an intermediate code corresponding to a rectangle;

Fig. 19 is a schematic view showing the meaning of each of bits composing a texture;

Fig. 20 is a schematic view showing the structure of an intermediate code corresponding to a polygon;

Fig. 21 is a schematic view showing the structure of an intermediate code corresponding to a circle or a circular arc;

Fig. 22 is a schematic view showing the structure of an intermediate code corresponding to blink;

Fig. 23 is a schematic view showing the configuration of software;

Fig. 24 is a block diagram functionally showing the configuration of a conventional FM subcarrier data broadcasting receiver;

Fig. 25 is a schematic view showing the configuration of software of the conventional FM subcarrier data broadcasting receiver; and

Fig. 26 is a flow chart showing the procedure for processing in a case where data constituting a reconstructed program are displayed in the conventional FM subcarrier data broadcasting receiver.

Best Mode for Carrying Out the Invention

Referring now to Figs. 1 to 23, embodiments of the present invention will be described.

Fig. 1 illustrates the electrical configuration of an FM subcarrier data broadcasting receiver.

A voice signal is reproduced through a stereo demodulating circuit 12, an amplifier 3, and a speaker 4 after its high-frequency component is removed by an LPF (Low Pass Filter) from an output signal of an FM tuner 1.

On the other hand, a voice component and a noise component are removed by a band-pass filter (BPF) 5 from the output signal of the FM tuner 1, so that a multiple signal (an LMSK signal) is extracted. The extracted multiple signal is fed to an LMSK demodulating and error correcting circuit 6.

The LMSK demodulating and error correcting circuit 6 subjects the fed multiple signal to LMSK demodulation as well as synchronous detection and error correction processing, to output packet data to a CPU 7.

The CPU 7 reconstructs data for each program on the basis of the packet data fed from the LMSK demodulating and error correcting circuit 6, to store the data in a RAM 9. A ROM 8 stores a program and the like of the CPU 7.

When a user operates an operating unit 11 such as a remote controller, to select a program, the CPU 7 decodes data constituting pages of the selected program, and displays the decoded data on a display

device 10 such as a liquid crystal display.

Fig. 2 shows the procedure for processing in a case where data constituting a reconstructed program are displayed.

When a drawing page is designated (step 1), an eight bit code in one display unit (for example, one character unit) constituting the page is decoded according to an eight bit coding system (step 2), and an intermediate code which can be decoded even if specification peculiar to FM subcarrier data broadcasting are not understood is generated (step 3). The details of the intermediate code will be described later.

When such processing is repeated for each data in one display unit, and all data constituting the page are subjected to the processing (YES at step 4), the program proceeds to the step 5.

At the step 5, the intermediate code is decoded, to obtain a character code or the like. A drawing pattern corresponding to the obtained character code or the like is obtained (step 6). The obtained drawing pattern is subjected to processing such as enlargement, underline addition, and inversion (step 7). The drawing pattern obtained after the processing is drawn in a designated position of the

display device (step 8).

When the processing at the steps 5 to 8 is repeatedly performed with respect to all intermediate codes, to complete drawing corresponding to one page (YES at step 9), the program is returned to the step 1.

The intermediate codes will be described by being divided into intermediate codes corresponding to instructions other than geometric drawing instructions and intermediate codes corresponding to the geometric drawing instructions.

[1] Description of Intermediate Codes Corresponding to Instructions Other Than Geometric Drawing Instructions

The types of intermediate codes (command types) corresponding to the instructions other than the geometric drawing instructions include the following 14 types, for example:

(1) a character (having no attribute)

(2) a character (having attributes)

(3) a one-layer photographic (having no attribute)

(4) a one-layer photographic (having an attribute)

(5) DRCS [Dynamically redefinable character

sets : an external character code set] (having no attribute)

(6) DRCS (having attributes)

(7) screen erasure

(8) TIME

(9) selective control information

(10) color change

(11) a color map

(12) a page attribute

(13) there is continuous data

(14) end

An example of codes representing the command types and the total number of bytes composing the intermediate code for each of the command types are shown in Table 1. In Table 1, the code representing the command type is indicated by HEX (a hexadecimal number).

Table 1

Code (HEX)	Command type	Number of bytes
F0	Character (having no attribute)	6
F1	Character (having attributes)	8
F2	One-layer photo (having no attribute)	8～52
F3	One-layer photo (having an attribute)	9～53
F4	DRCS (having no attribute)	14～52
F5	DRCS (having attributes)	16～54
F6	Screen erasure	2
F7	TIME	2
F8	Selective control information	43
F9	Color change	3
FA	Color map	5
⋮	⋮	⋮
FD	Page attribute	4
FE	There is continuous data	1
FF	End	1

Fig. 3 shows the structure of an intermediate code corresponding to the character (having no attribute) in the foregoing item (1).

The intermediate code is constituted by a command "F0" (one byte) representing the character (having no attribute), X coordinates (one byte) of

a display position, Y coordinates (one byte) of the display position, a character size (one byte), and a character code (2 bytes).

The character code includes a kanji character set (a JIS code), an alphabetic character set (one obtained by expanding a one-byte code to a two-byte code (for example, 29xxh)), a hiragana character set (one obtained by expanding a one-byte code to a two-byte code (for example, 2Axxh)), and a katakana character set (one obtained by expanding a one-byte code to a two-byte code (for example 2Bxxh)).

The character size is a one-byte code, and is determined as follows, for example:

0 : standard, 1 : medium-sized, 2 : small-sized, 3 : micro, 4 : two times the length, 5 : two times the width, 6 : two times the length and width, 7 : special 1, 8 : standard in European writing, 9 : unused, 10 : unused, 11 : unused, 12 : two times the length in European writing, 13 : two times the width in European writing, 14 : two times the length and width in European writing, and 15 : unused.

Fig. 4 illustrates the structure of an intermediate code corresponding to the character (having attributes) in the foregoing item (2).

The intermediate code is constituted by a

command "F1" (one byte) representing the character (having attributes), X-coordinates (one byte) of a display position, Y-coordinates (one byte) of the display position, a character size (one byte), a character code (two bytes), a font attribute (one byte), and a display attribute (one byte).

The font attribute is composed of eight bits (one byte), that is, b7 to b0, and is determined as follows:

The bit b7 represents the presence or absence of selective control cursor information, to indicate that there is no selective control cursor information if it is zero, while indicating that there is selective control cursor information if it is one. The bit b6 represents the presence or absence of an underline, to indicate that there is no underline if it is zero, while indicating that there is an underline if it is one. The bits b5 to b4 represent pattern polarities, to respectively represent "normal" if it is zero, "full inversion" if it is one, and "in-frame inversion" if it is two. The bits b3 to b0 represent enclosures, to respectively represent framing of a left side of a display section if b3 is one, an upper side of the display section if b2 is one, a right side of the

display section if b1 is one, and a lower side of the display section if b0 is one.

The display attribute is composed of eight bits (one byte), that is, b7 to b0, and is determined as follows:

The bits b7 to b6 are unused. The bits b5 to b4 represent a write mode, to respectively represent NEW writing (a mode of writing both a foreground color and a background color) if it is one, ON writing (a mode of writing only a foreground color) if it is one, and OFF writing (a mode of writing only a background color) if it is two. The bits b3 to b2 represent the presence or absence of flashing, to respectively indicate that there is no flashing if it is zero, indicate that there is positive phase flashing if it is one, and indicate that there is negative phase flashing if it is two. The bit b1 represents the presence or absence of conceal, to indicate that there is no conceal if it is zero, while indicating that there is conceal if it is one. The bit b0 represents non-spacing or spacing, to represent spacing if it is zero, while representing non-spacing if it is one.

Fig. 5 shows the structure of an intermediate code corresponding to the one-layer photographic

(having no attribute) in the foregoing item (3).

The intermediate code is constituted by a command "F2" (one byte) representing the one-layer photographic (having no attribute), X-coordinates (one byte) of a display position, Y-coordinates (one byte) of the display position, the size (one byte), and data. The type of size includes three types, that is, 4 × 4, 8 × 12, and 16 ×24. The number of bytes composing the data includes three types, that is, 4 bytes, 12 bytes and 48 bytes depending on the type of size.

Fig. 6 shows the structure of an intermediate code corresponding to the one-layer photographic (having an attribute) in the foregoing item (4).

The intermediate code is constituted by a command "F3" (one byte) representing the one-layer photographic (having an attribute), X-coordinates (one byte) of a display position, Y-coordinates (one byte) of the display position, a size (one byte), data (4, 12 or 48 bytes), and a display attribute (one byte).

Fig. 7 shows the structure of an intermediate code corresponding to the DRCS (having no attribute) in the foregoing item (5).

The intermediate code is constituted by a

command "F4" (one byte) representing the DRCS (having no attribute), X-coordinates (one byte) of a display position, Y-coordinates (one byte) of the display position, a character size (one byte), and data (10 to 48 bytes).

Fig. 8 shows the structure of an intermediate code corresponding to the DRCS (having attributes) in the foregoing item (6).

The intermediate code is constituted by a command "F5" (one byte) representing the DRCS (having attributes), X-coordinates (one byte) of a display position, Y-coordinates (one byte) of the display position, a character size (one byte), data (10 to 48 bytes), a font attribute (one byte), and a display attribute (one byte).

Fig. 9 shows the structure of an intermediate code corresponding to the screen erasure in the foregoing item (7).

The intermediate code is constituted by a command "F6" (one byte) representing the screen erasure, and a screen erasure area (one byte). The screen erasure area represents the whole area if it is zero, a header area if it is one, and a text area if it is two.

Fig. 10 shows the structure of an intermediate

code corresponding to the TIME in the foregoing item (8).

The intermediate code is a code for designating a period during which processing is interrupted, and is constituted by a command "F7" (one byte) representing the TIME, and the period during which processing is interrupted (one byte). The period during which processing is interrupted is 8-bit data, and a designated value × 0.1 (sec) is the period during which processing is interrupted.

Fig. 11 shows the structure of an intermediate code corresponding to the selective control information in the foregoing item (9).

The intermediate code is constituted by a command "F8" (one byte) representing the selective control information, and control information indicating what service identification (SI : destination service identification) is to be selected, what program (PROG : a destination program number) of the selected service identification is to be selected, and what page (PAGE : a destination page number) of the selected program is to be displayed when key entry is provided.

14 sets of control information are provided in correspondence with 14 types of key-in numbers, that

is, 0 to 9 and a to d in this example. When there is no selective control, SI = FFh.

Fig. 12 shows the structure of an intermediate code corresponding to the color change in the foregoing item (10).

The intermediate code is constituted by a command "F9" (one byte) representing the color change, a foreground color after the change (one byte), and a background color (one byte). The foreground color and the background color are respectively indicated by color map addresses {a color pallet (upper), and CMLA (a color map low address)(lower)}.

Fig. 13 shows the structure of an intermediate code corresponding to the color map in the foregoing item (11).

The intermediate code is constituted by a command "FA" (one byte) representing the color map, a color map address (one byte) of the color map to be rewritten, and color values (respective one bytes of R, G, and B). The intermediate code means that a header raster color is changed when the color map address is 80 h, and means that a raster color is changed when the color map address is 81 h.

Fig. 14 shows the structure of an intermediate

code corresponding to the page attribute in the foregoing item (12).

The intermediate code is constituted by a command "FD" (one byte) representing the page attribute, a presenting function (one byte), an information type (upper four bits), a display format (lower four bits), a header raster color (upper four bits), and a raster color (lower four bits). The presenting function is composed of 8 bits, and represents the type of sign included in a page and a function of presenting the sign. The contents of respective bits b1 to b8 composing the presenting function are shown in Table 2.

Table 2

Bit position	Contents of presentation
b1	"1" when a character (JIS level-1 and level-2 characters, DRCS, Mosaic, an addition sign) is included
b2	"1" when a graphic (a one-layer photographic, a geometric) is included
b3	"1" when transparent data 1 (traffic information data) is included
b4	"1" when text synthesis is possible
b5	Undefined
b6	Undefined
b7	"1" in case of a batch program, and "0" in case of a non-batch program
b8	"1" in case of a complementary program of voice broadcasting, and "0" in case of an independent program

The information type is information for identifying the information type of the contents of a program, and takes values from 0 (information type undesignated) to 15 (information type 15). A display format designates a display mode, and takes values from 0 (format 0) to 4 (format 4), and a value 15 (free format). The header raster color and the raster color are respectively composed of 4 bits,

and are indicated by CMLA.

An intermediate code corresponding to "there is continuous data" in the foregoing item (13) and an intermediate code corresponding to the end in the foregoing item (14) are respectively composed of a one-byte code respectively comprising corresponding commands "FE" and "FF".

[2] Description of Intermediate Codes Corresponding to Geometric Drawing Instructions

In a geometric drawing instruction, a starting point is given by the current drawing point or a designated point, and an end point is given by relative coordinates or absolute coordinates from the point. A method of moving a drawing point after executing a drawing instruction differs depending on the drawing instruction.

Specifically, in a drawing instruction "LINE" corresponding to a line and a drawing instruction "ARC" corresponding to a circle or a circular arc, a drawing point after executing the drawing instruction is moved to its end point. In a drawing instruction "RECT" corresponding to a rectangle, a drawing point after executing the drawing instruction is moved only in the X direction. In a drawing instruction "POINT" corresponding to a

point and a drawing instruction "POLY" corresponding to a polygon, a drawing point after executing the drawing instruction is not changed because its starting point and its end point coincide with each other.

In an intermediate code corresponding to the geometric drawing instruction, coordinates are unified into absolute coordinates. Specifically, the origin of coordinates is determined at an upper left vertex O of a display area e of a screen which is constituted by a header sentence display area e1 and a text display area e2, and a coordinate value is increased to the right and downward. Such a coordinate system is the same as a coordinate system for representing a display position in an intermediate code corresponding to a character, an intermediate code corresponding to a one-layer photographic, and an intermediate code corresponding to an external character code set.

The respective coordinates of X and Y representing the display position are represented by one byte in the intermediate code corresponding to the character, the intermediate code corresponding to the one-layer photographic, and the intermediate code corresponding to the external

character code set, while being represented by two-byte integers with signs in the intermediate code corresponding to the geometric drawing instruction.

The reason why the respective coordinates of X and Y are represented by two bytes in the intermediate code corresponding to the geometric drawing instruction is that coordinates outside the range of the display area e may, in some cases, be designated in the geometric drawing instruction, and may not be able to be represented by one byte in such a case. In the intermediate code corresponding to the geometric drawing instruction, the ranges of the X and Y coordinates are ranges indicated by the following expressions (1):

$$-256 \leqq X \leqq 512$$

$$-308 \leqq Y \leqq 460 \qquad \cdots (1)$$

Furthermore, in the intermediate codes corresponding to geometric drawing instructions, the necessity of managing the drawing point is eliminated by giving starting points to all the geometric drawing instructions. Correspondingly, the number of the types of geometric drawing instructions is reduced.

Table 3 shows the relationship between the type

of each of the geometric drawing instructions and the command type in the intermediate code corresponding to the geometric drawing instruction.

Table 3

Type of drawing instruction	Command in intermediate code
POINT SET ABS POINT SET REL	Not appear as an intermediate code
POINT ABS POINT REL	Point
LINE ABS LINE REL SET & LINE ABS SET & LINE REL	Line
ARC OUTLINED SET & ARC OUTLINED	Circle, circular arc (outlined)
ARC FILLED SET & ARC FILLED	Circle, circular arc (filled)
RECT OUTLINED SET & RECT OUTLINED	Rectangle (outlined)
RECT FILLED SET & RECT FILLED	Rectangle (filled)
POLY OUTLINED SET & POLY OUTLINED	Polygon (outlined)
POLY FILLED SET & POLY FILLED	Polygon (filled)

Table 4 shows the type of each of the geometric drawing instructions relating to a point (POINT) and its operation.

Table 4

Type	Operation
POINT SET ABS (one multi-valued operand)	A drawing point is set at an absolute coordinate value designated by an operand, but is not drawn.
POINT SET REL (one multi-valued operand)	A relative coordinate value designated by an operand is added to a coordinate value of the current drawing point to set a new drawing point, but the drawing point is not drawn.
POINT ABS (one multi-valued operand)	A drawing point is set at an absolute coordinate value designated by an operand, and is drawn in a foreground color by a point of the size of a logical pixel.
POINT REL (one multi-valued operand)	A drawing point is set at a relative coordinate value from the current drawing point designated by an operand, and is drawn in a background color by a point of the size of a logical pixel.

Table 5 shows the type of each of the geometric drawing instructions relating to a line (LINE) and its operation.

Table 5

Type	Operation
LINE ABS (one multi-valued operand)	The current drawing point is taken as a starting point, and an end point is designated at an absolute·coordinate value by a multi-valued operand.
LINE REL (one multi-valued operand)	The current drawing point is taken as a starting point, and an end point is designated at a relative coordinate value from the starting point by a multi-valued operand.
SET & LINE ABS (two multi-valued operands)	A starting point and an end point are respectively designated at absolute coordinate values by first and second multi-valued operands.
SET & LINE REL (two multi-valued operands)	A starting point is designated at an absolute coordinate value by a first multi-valued operand. An end point is designated at a relative.coordinate value from the starting point by a second multi-valued operand.

Table 6 shows the type of each of the geometric drawing instructions relating to a circle or a circular arc (ARC) and its operation.

Table 6

Type	Operation
ARC OUTLINED (two multi-valued operands)	A starting point is taken as the current drawing point, an intermediate point and an end point are respectively designated by a first operand and a second operand, to draw a circular arc or a circle in a color and a line texture currently designated.
ARC FILLED (two multi-valued operands)	An area composed of a circular arc and a chord by ARC OUTLINED and its inside is filled in a color and a texture pattern designated.
SET & ARC OUTLINED (three multi-valued operands)	A starting point, an intermediate point, and an end point are respectively designated by a first operand, a second operand, and a third operand, to draw a circular arc or a circle in a color and a line texture currently designated.
SET & ARC FILLED (three multi-valued operands)	An area composed of a circular arc and a chord by SET & ARC OUTLINED and its inside is filled in a color and a texture pattern designated.

Table 7 shows the type of each of the geometric drawing instructions relating to a rectangle (RECT) and its operation.

Table 7

Type	Operation
RECT OUTLINED (one multi-valued operand)	A starting point is taken as the current drawing point, the width and the height are designated by an operand, and four sides of a square are drawn in a color and a line texture designated.
RECT FILLED (one multi-valued operand)	An area composed of a square by RECT OUTLINED and its inside is filled in a color and a texture pattern designated.
SET & RECT OUTLINED (two multi-valued operands)	A starting point and the width and the height of a square are respectively designated by a first operand and a second operand, to draw the four sides of the square in a color and a line texture designated.
SET & RECT FILLED (two multi-valued operands)	An area composed of a square by SET & RECT OUTLINED and its inside is filled in a color and a texture pattern designated.

Table 8 shows the type of each of the geometric drawing instructions relating to a polygon (POLY) and its operation.

Table 8

Type	Operation
POLY OUTLINED	A starting point is taken as the current drawing point, coordinates of each of vertexes are designated by a multi-valued operand, and each of the sides of a polygon is drawn in a color and a line texture designated.
POLY FILLED	An area composed of a polygon by POLY OUTLINED and its inside is filled in a color and a texture pattern designated.
SET & POLY OUTLINED	A starting point and coordinates of each of vertexes are respectively designated by a first multi-valued operand and continuous multi-valued operands, to draw each of the sides of a polygon in a color and a line texture designated.
SET & POLY FILLED	An area composed of a polygon by SET & POLY OUTLINED and its inside is filled in a color and a texture pattern designated.

The types of intermediate codes (the command types) corresponding to the geometric drawing instructions are broadly classified into the following six types:

(1) a point

(2) a straight line

(3) a rectangle

(4) a polygon

(5) a circle or a circular arc

(6) blink

Examples of a code representing each of the command types and the total number of bytes composing an intermediate code for the command type are shown in Table 9.

Table 9

Code (HEX)	Command type	Number of bytes
E0	Point	7
E1	Straight line	12
E2	Rectangle	12
E3	Polygon	17～
E4	Circle or circular arc	16
E5	Blink	6

Fig. 16 shows the structure of an intermediate code corresponding to the point in the foregoing item (1).

The intermediate code is constituted by a command "E0" (one byte) representing the point, a

pixel size dx (one byte) in the X direction, a pixel size dy (one byte) in the Y direction, X-coordinates (two bytes) of a display position, and Y-coordinates (two bytes) of the display position. The pixel sizes dx and dy correspond to the size of a pen point in a case where it is assumed that drawing is performed with a pen.

Fig. 17 shows the structure of an intermediate code corresponding to the straight line in the foregoing item (2).

The intermediate code is constituted by a command "E1" (one byte) representing the straight line, a line texture (one byte) representing the type of line, a pixel size dx (one byte) in the X direction, a pixel size dy (one byte) in the Y direction, X-coordinates (two bytes) of a starting point, Y-coordinates (two bytes) of the starting point, X-coordinates (two bytes) of an end point, and Y-coordinates (two bytes) of the end point.

Fig. 18 shows the structure of an intermediate code corresponding to the rectangle in the foregoing item (3).

The intermediate code is constituted by a command "E2" (one byte) representing the rectangle, a texture (one byte) representing the type of

texture, a pixel size dx (one byte) in the X direction, a pixel size dy (one byte) in the Y direction, upper left X-coordinates (two bytes), and upper left Y-coordinates (two bytes), lower right X-coordinates (two bytes), and lower right Y-coordinates (two bytes).

The texture is composed of eight bits (one byte), that is, b8 to b1, and the meaning of each of the bits is determined as shown in Fig. 19.

Specifically, the bit 8 is used for distinguishing outlined display and filled display. The bit b7 is used for designating a color mode. The bit b6 is not used. The bits b5 and b4 are used for designating a texture pattern in the case of the filled display. The bit 3 is used for specifying whether or not shade is given (highlight) in the case of the filled display. The bits b2 and b1 are used for designating the type of line (a line texture) in the case of the outlined display.

Fig. 20 shows the structure of an intermediate code corresponding to the polygon in the foregoing item (4).

The intermediate code is constituted by a command "E3" (one byte) representing the polygon, the number of vertexes (one byte), a texture (one

byte) representing the type of texture, a pixel size dx (one byte) in the X direction, a pixel size dy (one byte) in the Y direction, X-coordinates (respective two bytes) of respective vertexes from the first point to the end point, and Y-coordinates (respective two bytes) of the respective vertexes from the first pinot to the end point.

Fig. 21 shows the structure of an intermediate code corresponding to the circle or the circular arc in the foregoing item (5).

The intermediate code is constituted by a command "E4" (one byte) representing the circle or the circular arc, a texture (one byte) representing the type of texture, a pixel size dx (one byte) in the X direction, a pixel size dy (one byte) in the Y direction, X-coordinates (two bytes) of a starting point, Y-coordinates (two bytes) of the starting point, X coordinates (two bytes) of an intermediate point, Y coordinates (two bytes) of the intermediate point, X coordinates (two types) of an end point, and Y coordinates (two bytes) of the end point.

Fig. 22 shows the structure of an intermediate code corresponding to the blink in the foregoing item (6).

The blink is a function of periodically

replacing a color value at a blink original color map address (a blink original color) with a color value at a blink destination color map address (a blink destination color), and executes operations of temporarily inserting the color value at the blink destination color map address into the blink original color map address at the time point where an on-interval is started, and returning the original color value before the on-interval to the blink original color map address at the time point where the on-interval is terminated (the time point where an off- interval is started).

The intermediate code is constituted by a command "E5" (one byte) representing the blink, an original color (one byte) of the blink, a destination color of the blink (one byte), an on-interval (one byte), an off-interval (one byte), and a start time (one byte).

Fig. 23 shows the configuration of software in the FM subcarrier data broadcasting receiver according to the above-mentioned embodiment.

The software comprises a receiving processing portion for acquiring data (layer 3 data) from the LMSK demodulating and error correcting circuit 6, a program reconstructing portion for classifying the

layer 3 data to construct a data group (layer 4 data), a program analyzing portion for decoding each of data units (layer 5 data) included in the data group and producing an intermediate code, and an intermediate code decoding portion for decoding the intermediate code to present information (layer 6 data). The intermediate code decoding portion manages a display processing portion and a drawing portion.

The display processing portion performs the drawing pattern acquisition processing at the step 6 shown in Fig. 2 and the pattern processing at the step 7. The drawing portion performs the drawing processing at the step 8 shown in Fig. 2.

In Fig. 23, the receiving processing portion, the program reconstructing portion, and the program analyzing portion are portions which relate to specifications peculiar to FM subcarrier data broadcasting. The intermediate code decoding portion, the display processing portion and the drawing portion are portions which do not relate to the specifications peculiar to FM subcarrier data broadcasting. That is, in the above-mentioned embodiment, the portions which relate to the specifications peculiar to FM subcarrier data broadcasting and the portions which do not relate

to the specifications peculiar to FM subcarrier data broadcasting are separated from each other in the configuration of the software. Therefore, in developing the software in the portions which do not relate to the specifications peculiar to FM subcarrier data broadcasting, that is, the intermediate code decoding portion, the display processing portion and the drawing portion, the development is possible even if the specifications peculiar to FM subcarrier data broadcasting are not understood. As a result, the FM subcarrier data broadcasting receiver becomes easy to develop.